Algorithms for Combining Belief Functions

Wagner Teixeira da Silva
Departamento de Ciência da Computação,
Universidade de Brasília,
Rio de Janeiro, Brazil

Ruy Luiz Milidiú
Departamento de Informática,
Pontifícia Universidade Católica do Rio de Janeiro,
Rio de Janeiro, Brazil

ABSTRACT

In general, combining Dempster–Shafer belief functions over a frame of n elements is a problem with exponential time complexity in n. This is a consequence of an exponential number of focal elements being generated when the focal elements of the belief functions being combined intersect. In order to avoid this undesirable behavior, we must impose some special structure on the focal sets. Our approach is to work with families of subsets that are closed under intersection. Hence, we present four polynomial time algorithms for combining some particular types of belief functions. In the first case, the case of Bayesian belief functions, we exploit the result that if any belief function is Bayesian, then the resulting belief function is also Bayesian. Thus the resulting belief function has no more than n focal elements. In the second case, the special case of polytomic belief functions over the same partition, we exploit the fact that no new focal elements are created. In the case of exact polytomic belief functions, we exploit the fact that the resulting belief function is also exact polytomic, with no more than n focal sets. Finally, in the case of hierarchical belief functions, we exploit the fact that the resulting belief function has the same hierarchy.

KEYWORDS: ***Dempster–Shafer theory, evidence theory, belief function, evidence combination, belief function combination, Bayesian belief function, polytomic belief function, hierarchical belief function, algorithm, computational complexity***

Address correspondence to Ruy Luiz Milidiú, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ, CEP 22453, Brazil.

International Journal of Approximate Reasoning 1992; 7:73–94

655 Avenue of the Americas, New York, NY 10010 0888-613X/92/$5.00

1. INTRODUCTION

Dempster-Shafer (DS) theory is a generalization of the Bayesian approach. This theory was first introduced by Dempster [1]. Later, it was reformulated by Shafer [2] in terms of finite discrete domains. This kind of domain is widespread in knowledge-based systems applications. DS theory is appropriate when the probabilistic model is incomplete. This is the case when it is impossible to determine some of the model's parameters, such as prior distributions or some of the conditional probabilities among the model's variables.

Let Θ be a frame of discernment, that is, a set of propositions where only one is true. Combining belief functions over Θ has a time complexity that is exponential on the size of Θ. Some polynomial time algorithms for combining special belief functions have already been proposed. The three most important of these algorithms were proposed by Barnett [3], Shafer and Logan [4], and Shafer et al. [5]. However, they do not explicitly compute the resulting belief function. These algorithms compute just the belief measures (degree of belief, basic probability assignment, etc.) for a few propositions. A similar computation for all possible propositions through these algorithms would be either impossible or require an exponential time complexity.

The algorithms proposed in this paper explicitly compute the resulting belief function and have polynomial time complexity on the input size. These algorithms are very useful because they fully and clearly express the resulting belief function. This is particularly convenient when, after the combination process, there is a decision process based on the belief measures of some subsets of the frame.

This paper is divided into seven sections. Section 2 summarizes the main topics of DS theory, such as belief function, basic probability assignment, plausibility, focal set, focal elements, belief functions over a frame of discernment Θ, belief functions defined over a partition Ω of Θ, special kinds of belief functions, and orthogonal sum. Section 3 describes a polynomial time algorithm that combines a Bayesian belief function with any other kind of belief function. Section 4 describes a polynomial time algorithm for the combination of polytomic belief functions defined over a common partition of Θ. Section 5 describes a polynomial time algorithm that computes the combination of exact polytomic belief functions defined over different partitions of Θ. Section 6 describes a polynomial time algorithm that combines hierarchical belief functions over a common hierarchy of Θ. The structure of this hierarchy is defined by a tree. Finally, Section 7 presents our general comments and conclusions on the construction of algorithms for computing the combination of belief functions.

2. DEMPSTER—SHAFER THEORY

Before introducing our four algorithms we first briefly review some of the basic concepts in DS theory. These concepts help to clarify the evidence combination procedures treated in this paper. A general introduction to this theory can be found in Shafer [2]. A more recent approach is presented by Shafer and Logan [4] and Shafer et al. [5].

First, we need to define some basic notation.

DEFINITION *Suppose that* $\Phi = \{A_1, A_2, \ldots, A_k\}$ *and let* Δ *denote a general associative binary operator. We define* $\Delta\Phi$ *by*

$$\Delta\Phi = A_1 \Delta A_2 \Delta \cdots \Delta A_k$$

EXAMPLES

1. Suppose that Φ is a set of real numbers and $\Delta = +$, or more generally $\Delta = \sum$; then

$$\sum \Phi = A_1 + A_2 + \cdots + A_k$$

2. Suppose that Φ is a family of sets and $\Delta = \cup$; then

$$\cup\Phi = A_1 \cup A_2 \cup \cdots \cup A_k$$

3. Suppose that Φ is a set of belief functions and $\Delta = \oplus$, where $\oplus$ is the orthogonal sum operator; then

$$\oplus\Phi = A_1 \oplus A_2 \oplus \cdots \oplus A_k$$

2.1. Belief Functions

Let Θ be a finite set of propositions. Assume that just one proposition is true, but we do not know which one. In this case, we call Θ a frame of discernment. Any subset A of Θ is a disjunction of all original propositions in A. Therefore, each $A \subset \Theta$ has the following properties

1. If there is $p \in A$ such that p is a true proposition, then A is also true.
2. If $A \subset B$ and A is true, then B is also true.

Our uncertainty on which subset of Θ contains the true original proposition can be expressed by means of a random set S of Θ. The value of Pr[S $= A$] measures one's belief in the hypothesis of proposition A being true, without accounting for the belief mass distributed among all subpropositions of A.

In the evidential approach of DS theory, the probability density function of the random set S is viewed as a function $m: 2^{\Theta} \to [0, 1]$, called a basic probability assignment (bpa). This function distributes the basic unit mass of belief among all subsets of Θ. Hence, we can define m by

$$m(A) = \Pr[\mathrm{S} = A]$$

for each subset $A \subset \Theta$. This function has the properties

$$m(\varnothing) = 0$$

and

$$\sum \{m(A) \mid A \subset \Theta\} = 1$$

where $m(A)$ measures the portion of belief that is specific to A and to none of its proper subsets.

How would the related measures $\Pr[\mathrm{S} \subset A]$ and $\Pr[\mathrm{S} \cap A \neq \varnothing]$ be interpreted? In DS theory jargon, these measures are respectively known as degree of belief in A, denoted by $\mathrm{Bel}(A)$, and degree of plausibility in A, denoted by $\mathrm{Pl}(A)$. The belief function Bel is also given by (2.1), and it attributes degrees of belief to every subset A of Θ:

$$\begin{aligned}\mathrm{Bel}(A) = \Pr[\mathrm{S} \subset A] &= \sum \{\Pr[\mathrm{S} = B] \mid B \subset A\} \\ &= \sum \{m(B) \mid B \subset A\} \end{aligned} \tag{2.1}$$

$\mathrm{Bel}(A)$ is interpreted as the total belief mass attributed to subset A. The Pl function given by (2.2) attributes degrees of plausibility to every subset of Θ:

$$\begin{aligned}\mathrm{Pl}(A) = \Pr[\mathrm{S} \cap A \neq \varnothing] &= \sum \{\Pr[\mathrm{S} = B] \mid B \cap A \neq \varnothing\} \\ &= \sum \{m(B) \mid B \cap A \neq \varnothing\} \\ &= 1 - \mathrm{Bel}(\overline{A}) \end{aligned} \tag{2.2}$$

$\mathrm{Pl}(A)$ is the belief mass that does not contradict the hypothesis of A containing a true proposition.

From (2.1) and (2.2) it is easy to see that $\mathrm{Bel}(A) \leq \mathrm{Pl}(A)$ and $\mathrm{Bel}(A) + \mathrm{Pl}(\overline{A}) = 1$. The doubt in A is given by $\mathrm{Bel}(\overline{A})$. Expression (2.2) relates $\mathrm{Pl}(A)$ to the doubt in A. If Θ is a frame of discernment, then a belief function $\mathrm{Bel}\colon 2^{\Theta} \to [0, 1]$ has the following basic properties:

1. $\mathrm{Bel}(\varnothing) = 0$.
2. $\mathrm{Bel}(\Theta) = 1$.
3. If $k > 0$ and $A_1, A_2, \ldots, A_k$ are nonempty subsets of Θ, then

$$\begin{aligned}&\mathrm{Bel}(A_1 \cup A_2 \cup \cdots \cup A_k) \\ &\quad = \sum \left\{(-1)^{|I|+1}\mathrm{Bel}(\cap \{A_i \mid i \in I\} \mid \varnothing \neq I \subset \{1, 2, \ldots, k\}\right\}\end{aligned}$$

The set $\Phi = \{A \subset \Theta \mid m(A) > 0\}$ is called the focal set of Bel, and its elements are the focal elements of Bel. The set $K = \cup\Phi$ is the core of Bel. It is easy to see that $\mathrm{Bel}(K) = 1$.

Let Ω be a partition of Θ, and let Ω^* denote the set consisting of all unions of elements of Ω. The set Ω^* is a σ-algebra of subsets of Θ, and Ω is a base for Ω^*. If Bel is a belief function over Θ, with focal set Φ satisfying $\Phi \subset \Omega^*$, then we also say that Bel is defined over the partition Ω. In this case,

$$\mathrm{Bel}_\Omega(\{P_1, P_2, \ldots, P_k\}) = \mathrm{Bel}(P_1 \cup P_2 \cup \cdots \cup P_k)$$

for all positive k and subset $\{P_1, P_2, \ldots, P_k\}$ of Ω, where Bel_Ω is the function Bel over Θ. This function can be interpreted as a belief function over Ω at the level of implementation (Shafer et al. [5]).

Next, we define some special types of belief functions. They show properties that are relevant to the construction of belief combination algorithms.

Support function. Let Bel be a belief function over Θ, let Φ be its focal set, and let K be its core. If $K \in \Phi$, then Bel is called a support function.

Vacuous function. Let Bel be a support function over Θ. If $\Phi = \{\Theta\}$, then Bel is called a vacuous belief function.

Simple support function. Let Bel be a support function over Θ, and let A be a nonempty proper subset or Θ. If $\Phi \subset \{A, \Theta\}$, then Bel is called a simple support function. The set A is called the focus of Bel.

Dichotomic function. Let Bel be a belief function over Θ, and Let A be a nonempty proper subset of Θ. If $\Phi \subset \{A, \bar{A}, \Theta\}$, then Bel is called a dichotomic belief function with dichotomy $\{A, \bar{A}\}$.

Polytomic belief function. Let Bel be a belief function over Θ, with focal set Φ. Also let $\Omega = \{A_1, A_2, \ldots, A_k\}$ be a partition of Θ. If Bel's focal set $\Phi \subset \{A_1, A_2, \ldots, A_K, \Theta\}$ then Bel is called a polytomic belief function defined over Ω.

Exact polytomic belief function. Let Bel be a polytomic belief. If $\Theta \notin \Phi$, then Bel is called an exact polytomic belief function defined over Ω.

Bayesian belief function. Let Bel be an exact polytomic belief function over Θ. If every focal element $A \in \Phi$ is atomic, that is, $|A| = 1$, then Bel is called a Bayesian belief function.

2.2. Evidence Combination

Bodies of evidence are not directly combinable, but the belief functions supported by them are. Hence, before combining the bodies of evidence, each one of them must be judged in order to obtain its corresponding belief function. Belief functions over the same frame and supported by independent bodies of evidence can be combined if the intersection of their cores is not empty. The resulting belief function is therefore supported by all the pooled bodies of evidence. The independent bodies of evidence, or interpretations of them, are combined through their belief function representations. The basic procedure to combine evidences is Dempster's rule described below. Let S_1 and S_2 be random sets of the frame Θ. Suppose that S_1 and S_2 are probabilistically independent and that $\Pr[S_1 \cap S_2 \neq \emptyset] > 0$. Let Bel_1 and Bel_2 be two belief functions with bpa's m_1 and m_2, respectively, defined for each $A \subset \Theta$ by

$$m_1(A) = \Pr[\mathrm{S}_1 = A]$$

and

$$m_2(A) = \Pr[\mathrm{S}_2 = A]$$

Therefore, the orthogonal sum Bel, defined by $\mathrm{Bel} = \mathrm{Bel}_1 \oplus \mathrm{Bel}_2$, has its bpa m defined for each $A \subset \Theta$ by

$$m(A) = \Pr[\mathrm{S}_1 \cap \mathrm{S}_2 = A \,|\, \mathrm{S}_1 \cap \mathrm{S}_2 \neq \emptyset]$$

When $R > 0$ this equation can be reformulated as

$$m(A) = R \sum \{m_1(A_1) m_2(A_2) \mid A_1 \cap A_2 = A,\ A_1 \in \Phi_1,\ A_2 \in \Phi_2\} \tag{2.3}$$

where R is defined by the equation

$$R^{-1} = \sum \{m_1(A_1) m_2(A_2) \mid A_1 \cap A_2 \neq \emptyset,\ A_1 \in \Phi_1,\ A_2 \in \Phi_2\}$$

where Φ_1 and Φ_2 the focal sets of Bel_1 and Bel_2, respectively. Equation (2.3) is the so-called Dempster's rule.

Let us assume that Θ is an abstract data type with union, intersection, membership, and contained operations. Suppose that the elements of the frame of discernment Θ are represented by $\{1, 2, \ldots, n\}$, where each $i \in \Theta$ labels a proposition. These numbers might, in practice, be indices into a symbol table where the actual propositions are stored. There always is a suitable data structure for Θ, such that any of these operations is performed in linear time $O(n)$.

2.3. Focal Sets

Let Bel, Bel_0, and Bel_1 be belief functions over the same frame of discernment Θ such that $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1$. Let also Φ, Φ_0, and Φ_1 be their respective focal sets. Hence, the focal set Φ is given by

$$\Phi = \Phi_0 \oplus \Phi_1 = \{A \cap B \mid A \in \Phi_0, B \in \Phi_1\}$$

Our goal is to keep the size of Φ as small as possible so that we can minimize the combination effort. Therefore, we must look for situations where the size of Φ is small even when Φ_0 and Φ_1 are large.

Next, we present some lemmas that have trivial proofs.

LEMMA 2.1 *If Ω is a partition of $\Theta, \Phi_0 \subset \Omega$, and $\Phi_1 \subset \Omega^*$, then $\Phi = \Phi_0 \oplus \Phi_1 \subset \Phi_0$.*

LEMMA 2.2 *If Ω is a partition of Θ and $\Phi, \Phi_1 \subset (\Omega \cup \{\Theta\})$, then $\Phi = \Phi_0 \oplus \Phi_1 \subset (\Omega \cup \{\Theta\})$.*

LEMMA 2.3 *If Ω_0 and Ω_1 are partitions of Θ, then $\Omega_0 \oplus \Omega_1 = \{A \cap B \mid A \in \Omega_0, B \in \Omega_1, A \cap B \neq \emptyset\}$ is also a partition of Θ.*

LEMMA 2.4 *If Ω_0, Ω_1 are partitions of $\Theta, \Phi_0 \subset \Omega_0$, and $\Phi_1 \subset \Omega_1$, then $\Phi_1 \oplus \Phi_1 \subset \Omega_0 \oplus \Omega_1$.*

LEMMA 2.5 *Let Ω be a partition of Θ, and $Y \subset \Omega^*$. Assume that for all distinct nonempty subsets A and B of Y, just one of the following possibilities is true: $A \subset B$, $B \subset A$, or $A \cap B = \emptyset$. Then Y is closed under intersection.*

LEMMA 2.6 *Let Ω and Y be defined as in Lemma 2.5. If $\Phi_0, \Phi_1 \subset Y$, then $\Phi = \Phi_0 \oplus \Phi_1 \subset Y$.*

3. BAYESIAN BELIEF FUNCTION

A Bayesian belief function is a particular kind of exact polytomic belief function, where all of the focal elements are atomic. When a Bayesian belief function is combined with any other belief function, this atomicity generates another Bayesian belief function.

3.1 Basic Results

THEOREM 3.1 *Suppose that Bel_0 and Bel_1 are combinable belief functions, and Bel_0 is a Bayesian belief function. Then the resulting belief function Bel, defined by $Bel = Bel_0 \oplus Bel_1$, is a Bayesian belief function too.*

Proof See Shafer [2]. ■

COROLLARY 3.1 *Suppose that $Bel_0, Bel_1, \ldots, Bel_k$ are combinable belief functions, with at least one of them being a Bayesian belief func-*

tion. Then the combined belief function $Bel = Bel_0 \oplus Bel_1 \oplus \cdots \oplus Bel_k$ *is a Bayesian belief function too.*

THEOREM 3.2 *Suppose that* Bel_0 *and* Bel_1 *are combinable belief functions over* Θ. *Let* Bel_0 *be a Bayesian belief function, and* Pl_1 *the plausibility function associated with* Bel_1. *Then, the bpa m associated with the belief function Bel, defined by* $Bel = Bel_0 \oplus Bel_1$, *is given by*

$$m(\{\theta\}) = Rm_0(\{\theta\})\,\mathrm{Pl}_1(\{\theta\})$$

for each $\theta \in \Theta$, *where* $R^{-1} = \sum\{m_0(\{\theta\})\,\mathrm{Pl}_1(\{\theta\}) \mid \theta \in \Theta\}$ *is the normalizing constant of Bel.*

Proof From Dempster's rule for the orthogonal sum it follows that

$$m(\{\theta\}) = R\sum \{m_0(A)m_1(B) \mid A \cap B = \{\theta\},\ A \in \Phi_0,\ B \in \Phi_1\}$$

Since all $A \in \Phi_0$ are atomic,

$$m(\{\theta\}) = Rm_0(\{\theta\})\sum \{m_1(B) \mid \{\theta\} \cap B \approx \varnothing,\ B \in \Phi_1\}$$

But the above sum is the same as expression (2.2) for Pl when A is replaced by $\{\theta\}$. Hence,

$$m(\{\theta\}) = Rm_0(\{\theta\})\,\mathrm{Pl}_1(\{\theta\})$$

for each $\theta \in \Theta$. ■

3.2. Algorithm

The high-level description of the *CombineBayes* algorithm just described appears as Algorithm 3.1. It is based on Theorems 3.1 and 3.3 and on Corollary 3.1. This algorithm combines a Bayesian belief function with any

```
Procedure CombineBayes(Mb,Φ,M) ;
{ Pl - real vector with n elements
  p  - integer variable
  A  - focal element (type set, subset of Θ)
}
  Begin
  For p:=1 to n do    Pl[p] := 0;
  For each A ∈ Φ do
    For each p∈A  do   Pl[p] := Pl[p] + M[order(A)];
  For p:=1 to n do    Mb[p] := Mb[p] × Pl[p];
  End ;
```

Algorithm 3.1. Combining a Bayesian belief function with any other belief function.

other belief function. The Bayesian belief function is represented by the vector M_b with n entries, whereas the other function is represented by it focal set Φ and by a real vector M, corresponding to its bpa m.

Suppose that the frame Θ is finite with $|\Theta| = n$. Then each $A \in \Phi$ is a subset of $\Theta = (1, 2, \ldots, n\}$. We can think of Φ and $A \in \Phi$ as lists at the level of implementation. We can take the bpa m as a real vector M indexed by the order of the focal element (sublist) A into the focal set (list) Φ. So we use $M[\text{order}(A)] = m(A)$, for each $A \in \Phi$. The real vector M is the representation of the bpa m restricted to Φ. However, we describe the algorithm using set notation by thinking of the sets Φ and A as abstract data types. To illustrate this algorithm, let us show an example.

EXAMPLE Let $\Theta = \{1, 2, 3, 4\}$ be a frame of discernment. Let Bel_0 be a Bayesian belief function over Θ, with focal set Φ_0 given by $\Phi_0 = \{\{1\}, \{2\}, \{3\}, \{4\}\}$ and bpa m_0 given by $M_b = (1/4, 1/4, 1/4, 1/4)$. In this case, $m_0(A) = 1/4$ for each $A \in \Phi_0$. Let Bel be another belief function over Θ whose focal set is $\Phi = \{\{1, 2\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$ and with bpa m given by $M = (3/6, 1/6, 2/6)$.

The orthogonal sum $\text{Bel}_0 \oplus \text{Bel}$ is calculated by first computing the plausibility function Pl associated to Bel, for each elementary proposition of Θ. After computing Pl we have its representation given by the real vector $\text{Pl} = (5/6, 6/6, 3/6, 3/6)$. Next, we must multiply, coordinate by coordinate, the vectors M_b and Pl to get the new Bayesian belief function. Numerically, we obtain

$$M_b \times \text{PL} \propto (5/24, 6/24, 3/24, 3/24)$$

If there are several belief functions to combine, then we call procedure *CombineBayes* iteratively until all belief functions have been combined. For each iteration it receives M_b, Φ, and m as inputs and returns M_b as the resulting Bayesian belief function.

After combining all given belief functions, the resulting Bayesian belief function must be normalized. The normalization procedure finds first the sum S of the elements of M_b and then divides each element of M_b by S. In the preceding example, the normalized Bayesian belief function is given by the vector

$$M_b = (5/17, 6/17, 3/17, 3/17).$$

3.3. Complexity Analysis

THEOREM 3.3 *Suppose that* $\text{Bel}_0, \text{Bel}_1, \ldots, \text{Bel}_k$ *are combinable belief functions over* Θ, *with focal sets of size* $f_0, f_1, \ldots, f_k$, *respectively. Let* $f = \max\{f_1, \ldots, f_k\}$. *Assume that* Bel_0 *is a Bayesian belief function. Then, the iterative application of the CombineBayes algorithm yields the combination of the belief functions in* $O(nkf)$ *time.*

Proof To combine $k + 1$ belief functions, the *Combine Bayes* algorithm has to be called k times. Hence, the computing time is proportional to k. Now we must know the complexity of the *CombineBayes* algorithm. The loop of line 1 iterates n times. Each iteration takes $O(1)$ time. Then its total time is $O(n)$. The nest of loops of lines 2 and 3 repeats $\sum\{ |A| \mid A \in \Phi \}$ times. However, $\sum\{ |A| \mid A \in \Phi \} \leq nf$, where $f = |\Phi|$, and each iteration takes $O(1)$ time. Therefore, the computing time of lines 2 and 3 is $O(nf)$. The time for the loop of line 4 has the same order as the one of line 1. Adding the computing time found at lines 1, 2 and 3, and 4 we obtain the computing time of the *CombineBayes* algorithm, which is $O(nf)$. If we take f as the greatest number of focal elements among the belief functions Bel_1 and Bel_k, then the time necessary to combine all the $k + 1$ belief functions is $O(nkf)$. ■

4. POLYTOMIES OVER THE SAME PARTITION

Combining polytomic belief functions over Θ, with each one of them defined over the same partition Ω and Θ, yields another function of the same kind. Suppose that Ω has f elements and there are $k + 1$ functions to be combined. In this section, we propose an algorithm that combines these polytomic belief functions defined over Ω in $O(kf)$ computing time.

4.1. Basic Results

THEOREM 4.1 *Let $\Omega = \{A_1, A_2, \ldots, A_f\}$ be a partition of Θ. Let* Bel_0 *and* Bel_1 *be two polytomic belief functions over Θ, both of them defined over the partition Ω. Then,* $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathit{Bel}_1$ *is also a polytomic belief function over Ω.*

Proof Taking Φ, Φ_0, and Φ_1 as focal sets of Bel, Bel_0, and Bel_1, respectively, we have by Lemma 2.2 that Bel is a polytomic belief function over the partition Ω. ■

COROLLARY 4.1 *Let Ω be a partition of Θ, and let* $\mathrm{Bel}_0, \mathrm{Bel}_1, \ldots, \mathrm{Bel}_k$ *be a list of polytomic belief functions over Θ, with each one of them defined over Ω. Then* $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1 \oplus \cdots \oplus \mathrm{Bel}_k$ *is also a polytomic belief function over Ω.*

4.2. Algorithm

Let $\Omega = \{A_1, A_2, \ldots, A_f\}$ be a partition of Θ, and let Bel be a polytomic belief function over the partition Ω. Then, for the focal set of Φ of Bel we have that $\Phi \subset \{A_1, A_2, \ldots, A_f, \Theta\}$. For computational convenience we join both Θ and the partition Ω in the vector $\mathbf{P} = (\Theta, A_1, A_2, \ldots, A_f)$. In this case, $P[0] = \Theta$ and $P[k] = A_k$ for $k = 1, 2,, \ldots, f$. We represent the bpa m of Bel by a real vector M with $f + 1$ positions. When $M[p] > 0$ then $P[p]$ is a focal element of Bel with $m(P[p]) = M[p]$, for each $p \in \{0, 1, \ldots, f\}$.

This structure allows us to combine polytomic belief functions in a simple way. Let Bel_0 and Bel_1 be two polytomic belief functions. We represent their bpa's by the real vectors M_0 and M_1, respectively. So $\text{Bel} = \text{Bel}_0 \oplus \text{Bel}_1$ has bpa M given by

$$M[p] = M_0[p](M_1[p] + M_1[0]) + M_0[0]M_1[p] \qquad (4.1)$$

for each $p \in \{0, 1, \ldots, f\}$. This expression is easily derived from the orthogonal sum. A high-level description of the *CommonPolytomy* algorithm just described is Algorithm 4.1. Its correctness is based on Theorem 4.1 and Corollary 4.1. This algorithm has inputs M_0 and M_1. They represent the bpa's of the belief functions Bel_0 and Bel_1, respectively. The output M represents the bpa of Bel. Using expression (4.1) for the computation of M, this algorithm does not yield a normalized function. This could be achieved by another procedure, which adds the elements of M to obtain the sum S. To get the normalized function, we need just to divide each element of M by S.

If there are $k + 1$ polytomic belief functions to be combined, we must call the *CommonPolytomy* algorithm k times. The normalization procedure is called only once, after all combinations have been done. This saves computing time.

Next, we present an example that illustrates the combination of two given polytomic belief functions.

```
Procedure CommonPolytomy (M0,M1,M) ;
{ M0 and M1 - Input real vectors with f+1 positions. They
             represent the bpa's of Bel0 and Bel1 .
  M - Output real vector with f+1 positions. It represents the
      bpa of Bel = Bel0 ⊕ Bel1.
  p - local integer variable. It is the pth proposition in P.
}
  Begin
  M[0] := M0[0] × M1[0] ;
  For p:=1 to f do
     M[p] := M0[p] × (M1[p] + M1[0]) + M0[0] × M1[p] ;

  End ;
```

Algorithm 4.1. Combining two polytomic belief functions.

EXAMPLE Let $P = (\Theta, A, B, C, D)$ be the vector of all possible focal elements of Bel_0 and Bel_1. Let $M_0 = (2/6, 0, 2/6, 3/6, 0)$ and $M_1 = (0, 1/3, 1/3, 0, 1/3)$ be their corresponding bpa's. After calling the *CommonPolytomy* algorithm, with the inputs M_0 and M_1, we get as a result the unnormalized bpa $M = (0, 2/18, 3/18, 0, 2/18)$. After normalization we get

$$M = (0, 2/7, 3/7, 2/7)$$

4.3. Complexity Analysis

THEOREM 4.2 *Let* $\Omega = \{A_1, A_2, \ldots, A_f\}$ *be a partition of* Θ. *Suppose that* $\mathrm{Bel}_0, \mathrm{Bel}_1, \ldots, \mathrm{Bel}_k$ *are combinable polytomic belief functions over the partition* Ω. *Then, the CommonPolytomy algorithm, called k times, combines these belief functions in* $O(kf)$ *computing time.*

Proof By examining Algorithm 4.1 we can easily access the computing time of the *CommonPolytomy* algorithm. Line 1 takes $O(1)$ time to compute Bel's bpa on Θ. The loop of lines 2–3 iterates f times, and each iteration takes $O(1)$ time. Therefore, the computation time of this loop is $O(f)$. Adding the computing time of lines 1 and 2–3, we obtain $O(f)$ computing time for the *CommonPolytomy* algorithm. As this algorithm is called k times for combining the $k + 1$ belief functions, then the total effort takes $O(kf)$ time. ∎

In addition, we observe that the space requirements of this algorithm are very small. Besides the three vectors M_0, M_1, and M, just an additional space of constant size is required.

5. EXACT POLYTOMY

An exact polytomic belief function is a particular case of polytomic belief functions. A polytomy is exact when Θ is not a focal element. In this case, the combination of exact polytomic belief functions always results in another exact polytomic belief function. Exploring this fact, we propose an algorithm to compute the combination of $k + 1$ belief functions in $O(kn)$ time.

5.1. Basic Results

THEOREM 5.1 *Suppose that* Ω_0 *and* Ω_1 *are partitions of* Θ, *and* Bel_0 *and* Bel_1 *are combinable exact polytomic belief functions defined over* Ω_0 *and* Ω_1, *respectively. Then,* $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1$ *is an exact polytomic belief function over* $\Omega_0 \oplus \Omega_1$.

Proof Theorem 5.1 is an immediate consequence of Lemma 2.4 by taking Φ_0, Φ_1, and $\Phi_0 \oplus \Phi_1$ as the focal sets of Bel_0, Bel_1, and Bel, respectively. ∎

COROLLARY 5.1 *Suppose that* Ω_i *is a partition of* Θ, *for* $i = 0, 1, \ldots, k$. *Let* $\mathrm{Bel}_0, \mathrm{Bel}_1, \ldots, \mathrm{Bel}_k$ *be a list of combinable exact polytomic belief functions defined over the partitions* $\Omega_0, \Omega_1, \ldots, \Omega_k$, *respectively. Then, the belief function* Bel, *defined by* $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1 \oplus \cdots \oplus \mathrm{Bel}_k$, *is an exact polytomic belief function defined over* $\Omega_0 \oplus \Omega_1 \oplus \cdots \oplus \Omega_k$.

5.2. Algorithm

There are at least two ways to represent an exact polytomic belief function Bel. First, directly by its focal set Φ and its bpa m. Second, by its inverted focal set Ξ and by the representation M of the bpa restricted to Φ.

Let $\Theta = \{1, 2, \ldots, n\}$, Ω a partition of Θ, and Bel defined over the partition Ω. Then, $\Phi \subset \Omega$ and Ξ is an integer vector with n positions. $\Xi[p]$ represents the position in Φ of the Bel's focal element that contains the pth proposition of Θ—here, we are considering Φ and Φ's focal elements as lists.

As an example, let $\Theta = \{1, 2, 3, 4\}$ and $\Omega = \{\{1\}, \{2\}, \{3, 4\}\}$. Suppose that Bel's focal set $\Phi = \{\{1\}, \{3, 4\}\}$ and Bel's bpa m is represented by $M = \{1/3, 2/3\}$ on Φ, that is, $m(\{1\}) = M[1]$ and $m(\{3, 4\}) = M[2]$. Then, the vector Ξ is given by $\Xi = (1, 0, 2, 2)$. $\Xi[1] = 1$ tells us that proposition 1 belongs to the first focal element of Bel, represented by $\{1\}$ in Φ. Similarly, $\Xi[3] = 2$ tells us that proposition 3 belongs to the second focal element of Bel, represented by $\{3, 4\}$ in Φ. However, $\Xi[2] = 0$ tells us that proposition 2 belongs to none of the focal elements in Φ.

The *ExactPolytomy* algorithm, described by Algorithm 5.1, implements the results in Theorems 5.1 and 5.2. It uses both representations of the exact polytomic belief functions. This algorithm combines two of these belief functions at a time. Receiving Bel_0 and Bel_1 as inputs, it combines them and returns as output its sum Bel defined by $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1$. Bel_0 is represented by its inverse focal set Ξ and by the representation M of its bpa. Bel_1 is represented by its focal set Φ and its bpa m. The resulting belief function Bel is represented by its inverse focal set Ξ, and its bpa's image M_r. This function is not normalized, so an additional procedure must normalize it.

EXAMPLE Let $\Theta = \{1, 2, 3, 4\}$. Let also Φ_0 be the focal set of Bel_0, with $\Phi_0 = \{\{2, 3, 5\}, \{1, 4\}\}$. The representation of the bpa m of Bel_0 is $M = (1/4, 3/4)$, and its inverse focal set Ξ is the vector $(2, 1, 1, 2, 1)$. The focal set Φ of Bel_1 is $\{\{2, 3, 4\}, \{1, 5\}\}$, and it bpa m is represented by $(1/3, 2/3)$. After calling *ExactPolytomy*(Ξ, M, Φ, m, Ξ_r, M_r), we obtain $E_r = (3, 1, 1, 2, 4)$ and $M_r = (1/12, 3/12, 6/12, 2/12)$. This result is already normalized because Bel_0 and Bel_1 have no conflicts between their supporting evidences.

```
Procedure ExactPolytomy (Ξ, M, Φ, m, Ξr, Mr) ;
{ Ξ[1..n] - integer vector containing the inverted representation
            of the focal set of Bel0
M[1..n] - real vector representing the bpa of Bel0
Φ - focal set of Bel1
m - bpa of Bel1
Ξr[1..n] - integer vector containing the inverted representation
            of the focal set of Bel
Mr[1..n] - real vector representing the bpa of Bel
nef - local; counter of focal elements of Bel
V[1..n] - global; integer vector. For A ∈ Φ, p ∈ A, and
        i=Ξ[p]>0, then V[i]=0 indicates that the i-th focal
        element of Φ0 was not intesected with A yet. Otherwise,
        V[1]>1.
p - local; integer variable; pth elementary proposition of Θ
}
 Begin
   nef :=0;
   For each A ∈ Φ do
   Begin
       For each p ∈ A do
          If (i:=Ξ[p])≠0, then
             If V[i]=0, then
                 Begin
                     nef := nef+1;
                     V[i] := nef;
                     Ξr[p] := nef;
                     Mr[nef] := m (A) × M[i];
                 End
             Else
                 Ξr[p] := V[i];
       For each p ∈ A do
          If (i:=Ξ[p])≠0, then V[i] := 0;
   End;
 End;
```

Algorithm 5.1. Combining two exact polytomic belief functions.

5.3. Complexity Analysis

THEOREM 5.2 *Given a list of* $k + 1$ *combinable exact polytomic functions over the same frame of discernment* $\Theta = \{1, 2, \ldots, n\}$, *then the ExactPolytomy algorithm, described by Algorithm* 5.1, *combines these functions in* $O(kn)$ *time.*

Proof The *ExactPolytomy* algorithm combines just two belief functions at a time. In order to combine all the functions, it must be called k times. So the total computing time grows linearly on k. Now we must estimate the computing time of the *ExactPolytomy* algorithm. In Algorithm 5.1, there is an external loop on line 2 and two internal loops on lines 4 and 15, respectively. The external loop of line 2 is repeated $|\Phi|$ times. For each $A \in \Phi$, the loops of lines 4 and 15 are repeated $|A|$ times. Consequently, these internal loops separately iterate a total of at most n times, since Φ is a subset of a partition of Θ and $\sum\{|A| \mid A \in \Phi\} \leq n$. Independently of what conditions are true in each one of these internal loops, each iteration takes $O(1)$ time. Adding the effort found in these loops, we obtain $O(n)$ computing time for the *ExactPolytomy* algorithm. Therefore, the k calls of *ExactPolytomy* take $O(kn)$■

6. HIERARCHICAL PROPOSITION SPACE

Let the tree $T = (V, E)$ be a hierarchy for the frame of discernment Θ, Bel a belief function over Θ, with its focal set Φ satisfying $\Phi \subset V$. Then, Bel is called a hierarchical belief function on T. The following example illustrates this concept.

EXAMPLE 6.1 Let $\Theta = \{1, 2, 3, 4, 5\}$, and $V = \{\Theta, \{1, 2\}, \{1\}, \{2\}, \{3, 4\}, \{3\}, \{4\}, \{5\}\}$. The corresponding tree structure is illustrated in Figure 6.1.

The combination of two hierarchical belief functions on T generates another hierarchical belief function on T. The naive algorithm combines two of these functions in $O(n^3)$ time, where n is the size of Θ.

Here, we propose an algorithm that combines two hierarchical belief functions in $O(n)$ computing time. Therefore, the combination of $k + 1$ belief functions of this kind takes $O(kn)$ time.

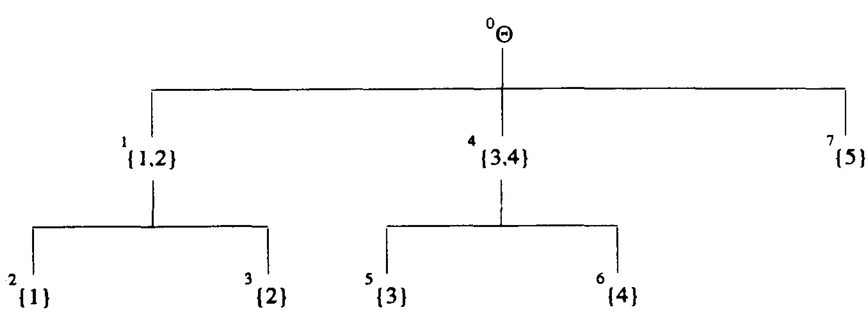

Figure 6.1. A hierarchy of $\Theta = \{1, 2, 3, 4, 5\}$.

6.1. Basic Results

DEFINITION 6.1 *Let T be a directed tree $T = (V, E)$ rooted on node Θ. Then, for each $A \in V$ we define the following concepts:*

- *$f(A)$ is the* father *of node A iff $(f(A), A) \in E$;*
- *$\Lambda(A)$ is the* ancestor set *of node A iff $\Lambda(A) = \Lambda(f(A)) \cup \{f(A)\}$, with $\Lambda(\Theta) = \varnothing$;*
- *$\Gamma(A) = \{B \mid (A, B) \in E\}$ is the* children set *of A;*
- *$\Delta(A) = \{B \mid A \in \Lambda(B)\}$ is the* descendants set *of node A;*
- *I is the* set of internal nodes *of T.*

DEFINITION 6.2 *Let Ω be a partition of Θ. The rooted tree $T = (V, E)$ is a* hierarchy *of Θ, iff*

- *The tree root is Θ,*
- *The tree leaves are the elements of Ω, and*
- *A is an internal node of T, $A = \bigcup\Gamma(A)$.*

DEFINITION 6.3 *Let* Bel *be a belief function over Θ, and let $T = (V, E)$ be a hierarchy of Θ. If the focal set Φ satisfies $\Phi \subset V$, then we say that* Bel *is* hierarchical *on T.*

THEOREM 6.2 *Suppose that $T = (V, E)$ is a hierarchy of Θ. Then V is closed under intersection.*

Proof Let A be any element of V. If A is not a leaf of T, then $\varnothing \neq \Gamma(A) \subset V$, and $A = \bigcup\Gamma(A)$. Hence, for any $A, B \in V$ we have the conditions of Lemma 2.5, and the proof is done. ■

THEOREM 6.2 *Let $T = (V, E)$ be a hierarchy of Θ. Let* Bel_0 *and* Bel_1 *he hierarchical belief functions on T. Then* $\text{Bel} = \text{Bel}_0 \oplus \text{Bel}_1$ *is also hierarchical on T.*

Proof Take Φ, Φ_0, and Φ_1 as the focal sets of Bel, Bel_0, and Bel_1, respectively. Since $\Phi = \Phi_0 \oplus \Phi_1$ and $\Phi_0, \Phi_1 \subset V$, it follows from Theorem 6.1 that $\Phi \subset V$. Therefore, Bel is hierarchical on T. ■

THEOREM 6.3 *Let* $\text{Bel}_0, \text{Bel}_1, \ldots, \text{Bel}_k$ *be combinable belief functions over Θ and defined on the same hierarchy $T = (V, E)$ of Θ. Then,* $\text{Bel} = \text{Bel}_0 \oplus \text{Bel}_1 \cdots \oplus \text{Bel}_k$ *is also hierarchical on T.*

The proof follows from Theorem 6.2 by using finite induction of k.

6.2. Algorithm

Suppose that Bel_0 and Bel_1 are combinable belief functions over Θ and that both are also hierarchical on the tree $T = (V, E)$, defined in Example 6.1. All possible intersections among the focal elements of Bel_0 and Bel_1 are illustrated in Table 6.1.

Table 6.1. All possible intersections of $\mathrm{Bel}_0 \oplus \mathrm{Bel}_1$

	m_1							
m_0	Θ	$\{1,2\}$	$\{1\}$	$\{2\}$	$\{3,4\}$	$\{3\}$	$\{4\}$	$\{5\}$
Θ	Θ	$\{1,2\}$	$\{1\}$	$\{2\}$	$\{3,4\}$	$\{3\}$	$\{4\}$	$\{5\}$
$\{1,2\}$	$\{1,2\}$	$\{1,2\}$	$\{1\}$	$\{2\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$\{1\}$	$\{1\}$	$\{1\}$	$\{1\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$\{2\}$	$\{2\}$	$\{2\}$	$\varnothing$	$\{2\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$\{3,4\}$	$\{3,4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{3,4\}$	$\{3\}$	$\{4\}$	$\varnothing$
$\{3\}$	$\{3\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{3\}$	$\{3\}$	$\varnothing$	$\varnothing$
$\{4\}$	$\{4\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{4\}$	$\varnothing$	$\{4\}$	$\varnothing$
$\{5\}$	$\{5\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\{5\}$

Figure 6.1 and Table 6.1 show that the intersection of A either with itself or with any other subset B, which is an ancestor of A, results in A. They also show that the intersection of A with any descendant B of A results in B. The remaining cases would result in empty sets.

Let $A = \{1\}$. In this case, $f(A) = \{1,2\}$. Observing the structure illustrated in Figure 6.1 and using the results in Table 6.1, we can compute the bpa m of $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1$ by the equation

$$m(A) \propto m_0(A)S_1(f(A)) + m_1(A)S_0(f(A)) + m_0(A)m_1(A) \quad (6.1)$$

where $S_i(f(A)) = m_i(\{1,2\}) + m_i(\Theta)$. Observe that this result in not normalized yet. We also observe the taking $A = \Theta$,

$$m(A) \propto m_0(A)m_1(A) \quad (6.2)$$

Equations (6.1) and (6.2) suggest a recursive procedure for the computation of m. Each node A of V should receive the values $S_i(f(A))$ from its father $f(A)$, for $i = 0, 1$. Next, it computes its bpa $m(A)$ from an equation similar to Eq. (6.1). The same procedure is called for each one of its children, with the arguments $S_i(A) = S_i(f(A)) + m_i(A)$, for $i = 0, 1$.

THEOREM 6.4 *Let* Bel_0 *and* Bel_1 *be hierarchical belief functions on the hierarchy* $T = (V, E)$ *of* Θ. *Suppose that* m_0 *and* m_1 *are the bpa's of* Bel_0 *and* Bel_1, *respectively. Then, the orthogonal sum* $\mathrm{Bel} = \mathrm{Bel}_0 \oplus \mathrm{Bel}_1$ *has bpa m given by*

$$m(A) = R[m_0(A)S_1(f(A)) + m_1(A)S_0(f(A)) + m_0(A)m_1(A)] \quad (6.3)$$

for each $A \in V$, *where* R *is the normalizing factor. The expression* *defining R, the recurrence relation for S_i, and its boundary conditions are given by the following equations.*

$$R^{-1} = \sum \{m_0(A)m_1(B) \mid A, B \in V, A \cap B \neq \varnothing\} \tag{6.4}$$

$$S_i(f(A)) = S_i(f(f(A))) + m_i(f(A)), \qquad i = 0, 1 \tag{6.5}$$

$$S_i(\varnothing) = 0, \qquad i = 0, 1 \tag{6.6}$$

$$m(\varnothing) = 0 \tag{6.7}$$

$$f(\Theta) = f(\varnothing) = \varnothing \tag{6.8}$$

Proof By Dempster's rule, each focal element of Bel_0 must intersect with all the focal elements of Bel_1 in order to yield the focal set of Bel. Since the focal sets of Bel_0 and Bel_1 are subsets of V, the resulting focal set Φ is contained in V. On the other hand, we know that for each $A \in V$ we have $A \cap B = A$ for each $B \in (\Lambda(A) \cup \{A\})$; $A \cap B = B$ for each $B \in \Delta(A)$; and $A \cap B = \varnothing$ otherwise.

For each $A \in V$, the orthogonal sum is obtained by the addition of the products $m_0(A)m_1(B)$, for $A \cap B = A$. This happens only when $B \in (\Lambda(A) \cup \{A\})$. The bpa m is given by

$$m(A) \propto m_0(A) \sum \{m_1(B) \mid B \in \Lambda(A)\} + m_1(A) \sum \{m_0(B) \mid B \in \Lambda(A)\} + m_0(A)m_1(A)$$

Defining S_i for $i = 0, 1$ by

$$S_i(f(A)) = \sum \{m_i(B) \mid B \in \Lambda(A)\}$$

and multiplying $m(A)$ by the normalizing factor R, we obtain Eq. (6.3).

The addition of the belief masses attributed to the focal elements of Bel before normalization gives the factor R^{-1}. By multiplying these belief masses by R^{-1}, we obtain the bpa m satisfying $\sum\{m(A)\} \mid A \in V\} = 1$, if we define R^{-1} through Eq. (6.4).

The sum $S_i(f(A))$ should be recursively computed by (6.5), since

$$\begin{aligned} S_i(f(A)) &= \sum \{m_i(B) \mid B \in \Lambda(A)\} \\ &= \sum \{m_i(B) \mid B \in (\Lambda(A) - \{f(A)\})\} + m_i(f(A)) \\ &= S_i(f(f(A))) + m_i(f(A)) \end{aligned}$$

for $i = 0, 1$ and $A \in V$.

Equations (6.6)–(6.8) set the necessary boundary conditions on S_i. ■

Algorithm *HierarchicalBelief* implements the results of Theorem 6.4. The resulting belief function is not normalized yet. Its high-level description appears as Algorithm 6.1. To combine two hierarchical belief functions Bel_0

Procedure *HierarchicalBelief* (x:**node**; S_0, S_1:**real**) ;
{ R - real; global; normalization factor
m_0 - global; bpa of Bel_0
m_1 - global; bpa of Bel_1
Mx_0 - local auxiliary variable
}
Begin
$Mx_0 := m_0$ (x);
m_0 (x) := m_0 (x) $\times S_1 + m_1$ (x) $\times S_0 + m_0$ (x) $\times m_1$ (x);
R := R + m_0 (x);
For each z $\in \Gamma(x)$ **do**
HierarchicalBelief (z, $S_0 + Mx_0$, $S_1 + m_1$ (x))
End

Algorithm 6.1. Combining two hierarchical belief functions.

and Bel_1 on $T = (V, E)$, we must call *HierarchicalBelief*(root_node, 0, 0), where root_node is the root of T. R, m_0, m_1 and the tree $T = (V, E)$ are global data structures. R is initialized with 0. The final value of R is used by the normalization procedure because the output of *HierarchicalBelief* is not normalized.

To combine $k + 1$ hierarchical belief functions on the same tree $T = (V, E)$, we must call *HierarchicalBelief* k times. Bel_0 will always be the resulting belief function of the preceding call and Bel_1 the next belief function on the list. Normalization is called once, just after all the functions have been combined. This saves computing effort.

6.3. Complexity Analysis

THEOREM 6.5 *Let Ω be a partition of Θ, and let Ω^* denote the set consisting of all the unions of elements of Ω. Assume that $T = (V, E)$ is a hierarchy with $V \subset \Omega^*$. The HierarchicalBelief algorithm described in Algorithm* 6.1 *combines two hierarchical belief functions on $T = (V, E)$ in $O(|\Omega|)$ time.*

Proof The *HierarchicalBelief* algorithm visits each node of T just once, because it traverses the tree recursively in preorder. In each node, it takes $O(1)$ time. Therefore, it takes $O(|V|)$ time to visit all the nodes of T. However, T is a hierarchy of Θ, and the elements of Ω are the leaves of T. Hence, $|V| \le 2|\Omega|$. Finally, we get that the total computing time of the *HierarchicalBelief* algorithm is $O(|\Omega|)$. ■

Table 6.2. The values computed by the *HierarchicalBelief* algorithm

v	x	m_0	S_0	m_1	S_1	m	R	m/R
0	Θ	0.1	—	0.2	—	0.02	0.02	0.04
1	$\{1,2\}$	0.3	0.1	0.4	0.2	0.22	0.24	0.39
2	$\{1\}$	0.2	0.4	—	0.6	0.12	0.36	0.21
3	$\{2\}$	—	0.4	0.1	0.6	0.04	0.40	0.07
4	$\{3,4\}$	0.2	0.1	0.3	0.2	0.13	0.53	0.22
5	$\{3\}$	—	0.3	—	0.5	—	0.53	—
6	$\{4\}$	—	0.3	—	0.5	—	0.53	—
7	$\{5\}$	0.2	0.1	—	0.2	0.04	0.57	0.07

The following example illustrates the preorder traversal performed by the *HierarchicalBelief* algorithm, and also the values computed at each visited node of T.

EXAMPLE 6.2 Let $T = (V, E)$ be the same hierarchy of Θ defined in Example 6.1. Let Bel_0 and Bel_1 be two hierarchical belief functions on T with bpa's m_0 and m_1 given in Table 6.2. The orthogonal sum $\text{Bel} = \text{Bel}_0 \oplus \text{Bel}_1$ computed by the *HierarchicalBelief* algorithm has an evolution suggested by Figures 6.2 and 6.3. Table 6.2 shows the following information for each node: (1) the values S_0 and S_1 that are passed as arguments to the algorithm; (2) the resulting bpa m; and (3) the factor R.

Figure 6.2 illustrates the bpa's m_0 and m_1 attached to each node of T. It also shows the sums S_0 and S_1 of the root's ancestors, which are both zeros before the *HierarchicalBelief* algorithm is called.

Figure 6.3 illustrates the preorder traversal when the 5th node is being visited. The number above each node suggests the traversal order. Observe

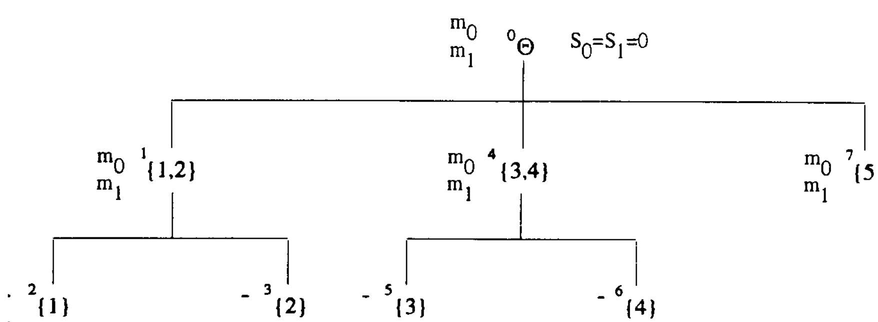

Figure 6.2. Computation of bpa for each node $x \in V$.

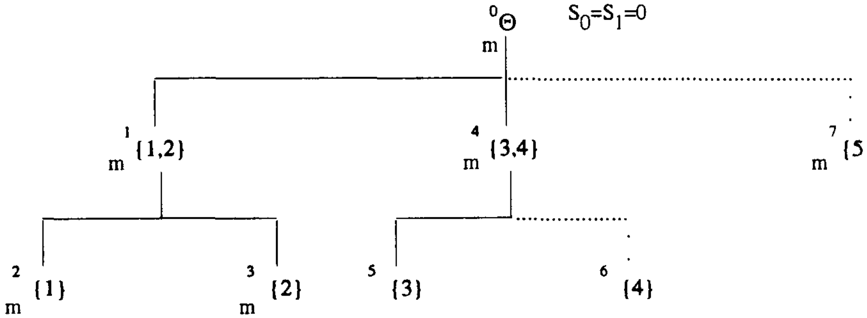

Figure 6.3. The propagation of S_0 and S_1, and the computation of R and m.

that when node $\{1\}$ is visited, it receives the values $S_0 = 0.4$ and $S_1 = 0.6$ from its father node $\{1, 2\}$. Since $m_0(\{1\}) = 0.2$ and $m_1(\{1\}) = 0$, then $m(\{1\}) = 0.12$, without normalization.

The normalization procedure can be recursive too. Figure 6.4 illustrates the preorder traversal made by such procedure. It must receive two arguments: the node and the factor R. Table 6.2 has the normalized bpa m in the column m/R.

7. CONCLUSION

Combining belief functions owes its complexity to the potential explosion in the number of focal elements of the resulting belief function. In order to avoid this undesirable behavior, we must somehow restrict the structure of the focal sets.

In this paper, we have shown some special cases of restrictions on the focal sets of belief functions that allow us to construct efficient algorithms for belief combination. Let n be the size of Θ. The following four cases were examined.

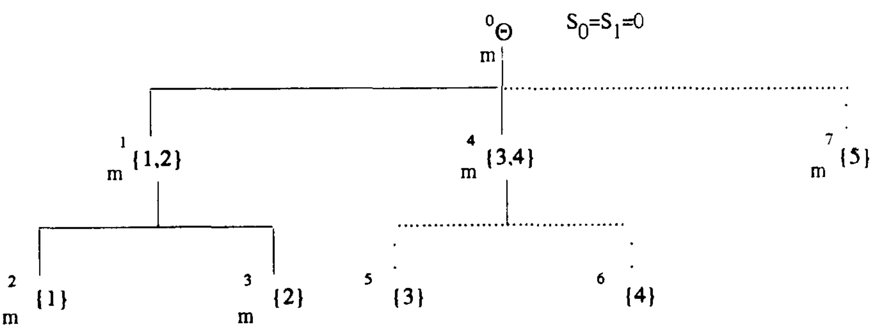

Figure 6.4. The normalization of m.

1. The Bayesian belief function, whose focal elements must be atomic, can be combined with any other belief functions. The resulting belief function is also Bayesian. Therefore, the size of its focal set is less than or equal to n.
2. The polytomic belief function that has as its focal elements the elements of a partition Ω of Θ, or the frame Θ itself, is defined over the partition Ω. When two or more polytomic belief functions are defined over the same partition Ω of Θ, then the combination of them is also a polytomic belief function defined over the same partition Ω. Therefore, the size of its focal set is no greater than n.
3. A polytomic belief function that does not have Θ as a focal element is called an exact polytomic belief function. Combining exact polytomic belief functions defined over different partitions of Θ yields another polytomic belief function also defined over a partition of Θ. Therefore, the size of its focal set is no greater than n.
4. For hierarchical belief functions, we have the underlying hierarchy of Θ defined by a tree $T = (V, E)$. The set of leaves of T corresponds to a partition of Θ. Suppose that the belief functions being combined have their focal sets as subsets of V. Then the focal set of the resulting belief function is also a subset of V. Since V is closed under intersection, the size of its focal set is no greater than $2n$. This result follows from the well-known relation $|V| \leq 2|\Omega|$.

In all four cases, the resulting belief function is completely specified. Therefore, any algorithm described in this paper can be used as a subprocedure of an inference system that models its uncertainty through belief functions.

References

1. Dempster, A. P., A generalization of Bayesian inference, *J. Roy. Stat. Soc. Ser. B* **30**, 205–247, 1968.

2. Shafer, G., *A Mathematical Theory of Evidence*, Princeton Univ. Press, Princeton, N.J., 1976.

3. Barnett, J. A., Computational methods for a mathematical theory of evidence, *Proceedings IJCAI-81*, Vancouver, BC, 868–875, 1981.

4 Shafer, G., and Logan, R., Implementing Dempster's rule for hierarchical evidence, *AI* **33**, 271–298, 1987.

5. Shafer, G., Shenoy, P., and Mellouli, K., Propagating belief functions in qualitative Markov trees, *Int. J. Approx. Reasoning* **1**, 349–400, 1987.